

# Gaining Insight into Femtosecond-scale CMOS Effects using FPGAs

**Kenneth M. Zick, Matthew French and Sen Li**

University of Southern California – Information Sciences Institute (USC ISI)  
Arlington, VA, USA, 22203 [www.isi.edu](http://www.isi.edu)  
Contact Author Email: [kzick@isi.edu](mailto:kzick@isi.edu)

**Abstract:** Reliability data and accurate compact models for new semiconductor processes are often very expensive to obtain. This paper considers low-cost collection of empirical data using COTS FPGAs and novel self-test. Hardware experiments using a 28 nm FPGA demonstrate isolation of small sets of transistors, detection of subtle aging, and measurement precision more than  $10\times$  finer (30–60 femtoseconds) than state of the art.

**Keywords:** Wear out; aging; NBTI; characterization; self-test; high precision; CMOS; FPGA.

## Introduction

Each semiconductor process technology brings a unique set of reliability characteristics and physical behaviors that are difficult for the government to accurately observe and model. Empirical reliability data from the foundry is typically inaccessible or insufficient for government applications. Ad hoc test chips can be designed and characterized, but at great cost. An additional possibility is the collection of extensive empirical data using COTS devices capable of self-test. We consider the use of FPGAs and novel characterization methods for this purpose. Specifically, we study whether small sets of transistors in FPGAs can be isolated, and whether high precision can be achieved such that subtle aging effects can be observed. We find that there are indeed cases in which individual transistors can be monitored, and that by taking careful measurements, effects such as NBTI can be monitored through low-cost highly-parallel self-test, at levels of precision  $10\times$  finer (30–60 femtoseconds) than state-of-the-art.

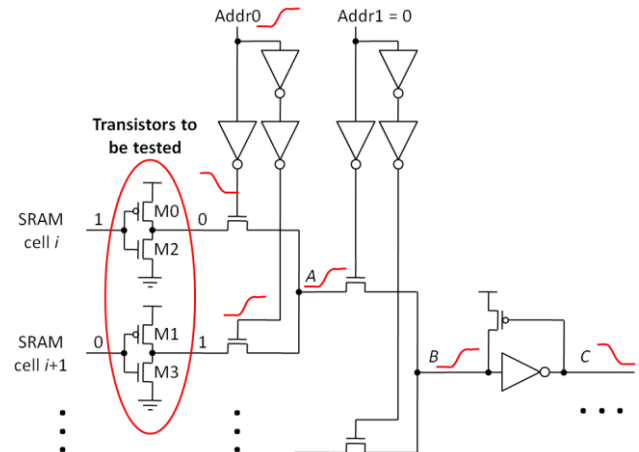
## Related Work

Most previous work involving delay characterization using FPGAs has involved relatively coarse precision. Some of the highest levels of precision achieved have been 0.61 ps [1] and 3.2 ps [2]. Aging in a 65 nm FPGA has been successfully tracked [3]. We explored femtosecond-scale characterization in a hardware security context [4]; that work is the basis for this paper. The idea of creating a reliability lab-on-a-chip using FPGAs has been recently considered by Pfiefer et al. [5]; the claimed resolution (not precision) is 0.1 ps. That approach characterizes ring oscillators implemented in a reconfigurable fabric. Though useful in some contexts, the paths involved are dominated by FPGA interconnect delays which are not always of interest or representative. Moreover, delays are measured

over a large number of transistors in the ring rather than a small number of isolated transistors.

## Proposed Approach

**Identifying Components to Monitor:** A major challenge in using FPGAs for low-level characterization is that the circuit design details are proprietary. Fortunately, information is available about typical transistor-level structures of SRAM-based lookup tables (LUTs), through patents and academic models such as [3]. LUTs are a particularly strong candidate for monitoring aging since their SRAM cells typically hold static data; certain transistors will be statically biased and potentially prone to bias-dependent effects. With many semiconductor technologies, transistors will degrade under certain gate-source bias, especially at high temperature; this bias temperature instability (BTI) effect shifts a transistor's threshold voltage and limits its saturation drain current. PMOS devices degrade under negative bias (NBTI) and NMOS under positive bias (PBTI). In some cases partial recovery occurs when stress is absent.



**Figure 1.** Portion of an SRAM-based lookup table circuit. Individual transistors can be monitored and compared with high-precision. A test case for transistor M1 is shown.

Fig. 1 shows a representative design of a portion of LUT circuitry, similar to [3]. SRAM cells (not shown) hold the static configuration data specifying the LUT function. A set of inverters drive the SRAM contents into a pass-gate multiplexor tree; we will refer to these inverters as the SRAM drivers, with transistors labeled M0:M3 in the figure. The polarity of the signals driven by the SRAM drivers is not documented; it could be true or complemented polarity with respect to the SRAM contents.

The SRAM driver transistors are statically biased; we hypothesize that these transistors might act as “canaries” for subtle degradation of the BTI type. Characterizing their switching speeds may not be practical, since their gates are controlled by the LUT contents which normally can only be changed during the reconfiguration process. There is another possibility—characterizing their drive strengths while they are turned on. The drivers are responsible for charging and discharging the nodes along the multiplexor tree (such as nodes *A* and *B*) when the LSB of the address bus (Addr0) switches. The delay through a path of this type can be measured, potentially providing information about the drive strength of the transistor of interest. Note that the saturation drain current depends on the threshold voltage [6], approximated by

$$I_{D,sat} \propto \left( V_{GS} - V_t - \frac{V_{D,sat}}{2} \right)^{\gamma}. \quad (1)$$

Therefore, a measured increase in delay can be an indication of degraded current capability and associated  $V_t$ , for instance caused by NBTI wear out.

A set of four FPGA configurations can be designed that allows characterization of the four types of transistors of interest. Similar to known methods of FPGA delay fault testing [7], the LUT can be configured with either of two alternating patterns, and the LSB of the address bus (Addr0) connected to the launch signal while the other address inputs are selectable per test. The launch signal can be driven with either a rising or falling edge to trigger a path. Subsequently, a single transistor under test will be responsible for pulling downstream nodes high or low through the NMOS pass gates. In the Fig. 1 example, nodes *A* and *B* can be charged up through M0 or M1, or discharged through M2 or M3. We refer to paths tested by Addr0 ↓ (↑) as “even” (odd) paths, and we refer to paths in which the LUT output rises (falls) as “rising” (“falling”) paths. A typical FPGA LUT may have 6 address inputs and 64 SRAM cells, and thus have 64 buffers containing 128 transistors of interest.

The paths from Addr0 to the LUT output depend not only on M0:M3 but also on the varying delays from Addr0 to the first level of pass gates, and on downstream NMOS pass gates and level restorers such as the one between nodes *B* and *C*. Delays through the downstream logic will be affected by dynamic activity on the address bus as well as data-dependent wear out. To isolate component shifts in M0:M3, relative comparisons are performed that subtract out the “noise” of the upstream and downstream logic. The measured parameters for each SRAM driver can be compared against those of the adjacent driver and others in the same neighborhood.

*Improvements to Test Architecture:* Measurements of component delays are subject to both random and systematic error. Random error is the unpredictable deviation in measured values caused by clock jitter, thermal

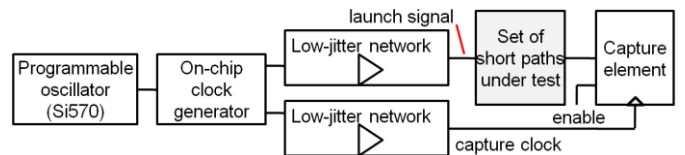
fluctuations, and many other types of noise. The amount of error dictates the precision of the measurement system. Random error can often be lessened through repeated measurements, but only partially and with a linear increase in test time.

In delay testing, a pair of test vectors that sensitize a path or node of interest is applied to the circuit under test and the result is captured by a sequential element, providing a yes/no answer per trial. Some methodologies prescribe repeating the process with different timing intervals between the launch signal and the capture signal; this is alternatively called launch-and-capture or clock sweeping. With FPGAs, testing has traditionally been performed on relatively long paths, representing a full clock phase or even a full clock period. This is appropriate for characterizing critical paths or detecting gross path delay faults, but for characterizing subtle aging effects, there is a need to isolate very short paths and detect very fine (sub-picosecond) phenomena.

For isolating short paths, we start with on-chip phase shifting. Using a single source such as an on-chip voltage controlled oscillator, two clocks are generated with a small phase shift. As an example, Xilinx Kintex-7 clock circuitry can generate phase shifts of roughly 13 ps. This enables characterization of paths nearly 100× shorter than typical (180-degree launch and capture is limited to paths longer than 1240 ps in [2]).

However, standard phase shifting is inadequate on its own, since the minimum phase increment (e.g. 13 ps) is orders of magnitude larger than the desired resolution and precision. For higher precision, we recommend the use of arbitrary phase shifting of launch and capture signals, via fine adjustments to the frequency of the reference clock feeding the above mentioned clock generator. To perform a sweep, the on-chip fractional phase shift is set to a fixed value (e.g. 1/56th of the voltage-controlled oscillator aka VCO period), and the reference clock is very finely adjusted with a separate programmable oscillator. Existing oscillators (such as the Si570 available on many Xilinx and Altera development boards) have resolution better than 0.09 parts per billion, readily supporting clock sweeps with femtosecond-scale step sizes.

The proposed test architecture is illustrated in Fig. 2. Further details regarding minimizing random and systematic error can be found in Zick et al. [4].



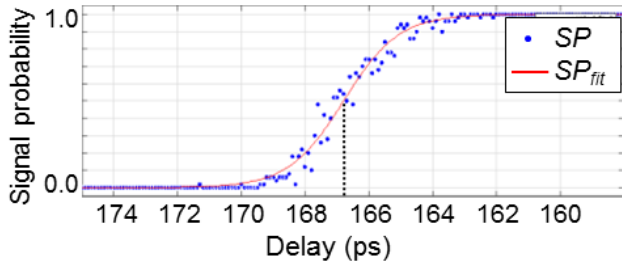
**Figure 2.** Overview of proposed test architecture

*Data Analysis Considerations:* The launch-and-capture characterization methodology provides a set of signal probabilities (*SP*) versus timing. Typically, a simple rule is

employed to convert the signal probability data to a path delay estimate. For instance, it is common to use the clock period at which  $SP$  is closest to 0.5 as the estimated delay [1]. Another rule is to use the lowest clock frequency at which the path fails at least once [8]. We refer to these rules as Nearest-to-Midpoint and First Fail, respectively. While these simple single-point rules have the advantage of being lightweight, they are prone to significant random error in the selected point, and are not suitable for the highest-precision characterization. We recommend using more of the data points in order to improve the precision. Assuming normally-distributed noise, the  $SP$  data is essentially the integral of a normal distribution. It is known that such a scenario can be modeled as a sigmoid function. Specifically,  $SP$  data can be fitted to a curve

$$SP_{fit} = \frac{1}{1 + e^{-A(x-B)}} \quad (2)$$

where  $x$  is the launch-to-capture interval,  $A$  is the slope, and  $B$  is the mid-point of the curve. The path delay can then be estimated from the curve simply by using the value of  $B$  (when  $x = B$ ,  $SP_{fit} = 0.5$ ). An example of a set of signal probabilities and the associated curve are shown in Fig. 3.



**Figure 3.** Estimating path timing more precisely by fitting signal probability data to a curve and finding the 50% point. In this case, the path delay is 166.800 ps;  $R^2 = 0.9987$ .

## Experimental Results

*Experimental Setup:* The base characterization platform is a Xilinx Kintex-7 FPGA manufactured in the TSMC 28HPL high-K metal gate process, mounted on a KC705 development board. The components under test are 2048 transistors associated with 1024 LUT SRAM cells in the style of Fig. 1. The transistors reside in 16 6-input LUTs located in slices X36Y179 to X39Y179, close to the on-chip XADC sensors which monitor voltage and temperature. Sixteen components are tested in parallel (one in each LUT). Each of the four types of transistors is tested with a unique FPGA configuration. The system design includes instrumentation, a MicroBlaze CPU, and a control program, all developed with Xilinx tools version 13.4. Curve fitting is performed using Eq. 3 and the `nlinfit` function in MATLAB.

*Overhead:* The time for each 16-LUT trial is measured to be roughly 1.5 ms, largely dictated by memory accesses between the MicroBlaze and the peripheral containing the launch and capture instruments. Additional overheads are

associated with changing the frequencies of the external oscillator (12 ms lock time) and on-chip clock generator, and the printing of data over a UART running at 460800 baud. In total, when collecting signal probability data over a range of 85 ps with 0.025 ps resolution, with 1 sample of size 50 at each timing point, the time is 5 minutes for each set of 512 paths.

*Precision:* To quantify the precision of the measurement approach, self-test was conducted multiple times consecutively. For each of the four types of paths, the test configuration was loaded, the set of 512 paths was tested over a span of 5 minutes, and the testing was repeated for a total of 3 measurements over 15 minutes. Ambient temperature was controlled to 25°C using a thermal chamber. The standard deviation of measurements of the same path turns out to be extremely small as shown in Table 1, averaging 0.063 ps for all 2048 paths. This level of repeatability is 10× finer than the measurements of a single path in [1], and roughly 50× finer than the repeatability of the estimates in [2]. When interleaving multiple measurements, the standard deviation is even lower, reaching 0.031 ps.

The fine degrees of precision achieved are due to a combination of test architecture, data analysis, and particulars of the implementation such as clock generator jitter performance. For the data analysis in particular, we can quantify the impact of the various strategies for mapping signal probabilities to path delay. We generated separate path delay estimates from the same exact signal probability data using the three strategies mentioned in section 4.3: the conventional Nearest-to-Midpoint rule, the First Fail rule, and our recommended Sigmoid Fit. Results show that Sigmoid Fit enables substantial improvements in precision, achieving levels 3.3× better than Nearest-to-Midpoint, as shown in Table 2. The First Fail rule was found to be highly susceptible to outliers as expected; Sigmoid Fit performs 8.2× better.

**Table 1.** Standard Deviation, Consecutive Measurements

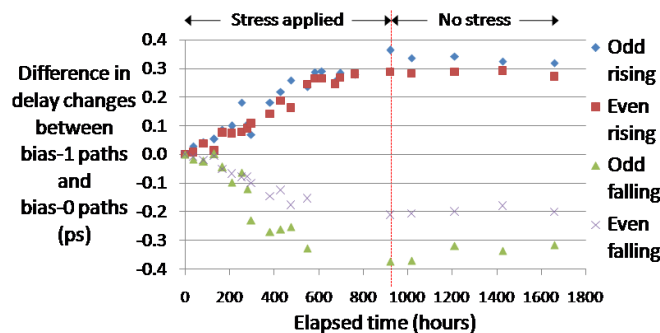
Path type	Ave. $\sigma$ for all paths (ps)	Worst case $\sigma$ (ps)
Even/rising	0.050	0.151
Even/falling	0.094	0.234
Odd/rising	0.050	0.176
Odd/falling	0.058	0.169

**Table 2.** Improvement in Measurement Precision through Curve Fitting (Sigmoid Fit)

Strategy for mapping data to delay estimate	Ave. $\sigma$ for all 2048 paths (ps)	Worst case $\sigma$ (ps)
First Fail	0.613	1.922
Nearest-to-Midpoint	0.205	1.097
Sigmoid Fit	0.063	0.234

**Detection of Component Delay Changes:** To test whether subtle component changes could be detected, the 2048 paths under test were statically stressed over a period of 922 hours with a random 1024-bit pattern in the associated SRAM cells, with occasional breaks for characterization. The chip junction temperature  $T_J$  was controlled using a reconfigurable heater circuit to accelerate aging. Temperatures were measured using the on-chip XADC sensor. The temperature schedule was 319 K for 82 hours; 358 K for 198 hours;  $\sim 388$  K for 642 hours. The wear out acceleration factors at these temperatures relative to room temperature (298 K) is 1.5–3.9 under the assumptions of 3. The launch signal was connected to the Addr0 input of each LUT and allowed to switch continuously during stress. After the 922 hours of stress, the device was left powered off for several weeks except for occasional characterization lasting  $\sim 1$  hour. All measurements were performed with  $T_J$  near 30°C.

We found that evidence of sub-picosecond, differential wear out is measurable, even after just days of stress. During stress, rising paths that had been biased at 1 incur an increasing slowdown relative to those biased at 0. Falling paths exhibit the opposite. Delay trends are plotted in Fig. 6. In just 8.25 days of operation at 85°C (between hours 82 and 280), a clear separation emerged for all four path types. For instance, at hour 280, the 271 even/rising paths biased at 1 had slowed by 0.1 ps more than the 241 even/rising paths biased at 0. Once the stress was completed and the device left powered off, the delays largely stabilized, validating that the degradation was caused by the stress and was semi-permanent.



**Figure 4.** Subtle differences are detected over time between bias-1 paths and bias-0 paths. Sub-picosecond effects are seen in all four path types.

## Conclusion

This research contributes methods for characterizing CMOS circuitry at low cost with astonishingly fine precision using COTS FPGAs and novel self-test. The approach can be extended to measuring thousands of paths in parallel. Whereas much previous work involves time scales of tens to hundreds of picoseconds, the enhanced methods reach tens of *femtoseconds*. High precision

requires special design and analysis methods; as one example, our curve fitting provided a  $3.3\times$  improvement, independent of other factors. Improving precision by 10–100 $\times$  opens up new possibilities for gaining empirical insight into the reliability of individual ICs and a particular semiconductor technology.

## Acknowledgements

The authors would like to acknowledge the support provided by Neil Steiner, Andrew Schmidt, and USC ISI Disruptive Electronics. This material is based upon work supported by the Defense Advanced Research Projects Agency (DARPA) under Contract No. HR001-11-C-0041. The views expressed are those of the authors and do not reflect the official policy or position of the Department of Defense or the U.S. Government. Distribution Statement “A” (Approved for Public Release, Distribution Unlimited).

## References

1. J.S.J. Wong, P. Sedcole, and P.Y.K. Cheung, “Self-characterization of combinatorial circuit delays in FPGAs,” *Int’l Conf. Field-Programmable Technology (FPT)*, pp. 17–23, 2007.
2. B. Gojman, S. Nalmela, N. Mehta, N. Howarth, and A. DeHon, “GROK-LAB: generating real on-chip knowledge for intra-cluster delays using timing extraction,” *Proc. Int’l Symp. Field Programmable Gate Arrays (FPGA)*, pp. 81–90, 2013.
3. E. Stott, J. Wong, P. Sedcole, and P.Y.K. Cheung, “Degradation in FPGAs: measurement and modelling,” *Proc. Int’l Symp. Field Programmable Gate Arrays (FPGA)*, pp. 229–238, 2010.
4. K. Zick, S. Li, and M. French, “High-precision self-characterization for the LUT burn-in information leakage threat,” *Int’l Conf. Field Programmable Logic and Applications*, 2014.
5. P. Pfeifer, B. Kaczer and Z. Pliva, “A reliability lab-on-chip using programmable arrays,” *IEEE Int’l Reliability Physics Symposium*, 2014.
6. A. DeHon and N. Mehta, “Exploiting partially defective LUTs: Why you don’t need perfect fabrication,” *Int’l Conf. Field-Programmable Technology (FPT)*, pp. 12–19, Dec. 2013.
7. P. Girard, O. Heron, S. Pravossoudovitch, and M. Renovell, “Defect analysis for delay-fault BIST in FPGAs,” *IEEE On-Line Testing Symposium*, pp. 124–128, 2003.
8. K. Xiao, X. Zhang, and M. Tehranipoor, “A clock sweeping technique for detecting hardware Trojans impacting circuits delay,” *IEEE Design & Test*, vol. 30, no. 2, pp. 26–34, April 2013.